

Nama : Devin Januar Siahaan

NIM : 24060121140158

Lab : PBO C1

PRAKTIKUM 9

Persistent Object

A. Menggunakan Persistent Object sebagai model basis data relasional

1. PersonDAO.java

```
/*
 * File : PersonDAO.java 31/05/23
 * Penulis : Devin Januar Siahaan - 24060121140158
 * Deskripsi : Interface untuk person access object
 */
public interface PersonDAO {
    public void savePerson(Person P) throws Exception;
}
```

2. Person.java

```
/**
 * File : Person.java 31/05/23
 * Penulis : Devin Januar Siahaan - 24060121140158
 * Deskripsi : Person database model
 */
public class Person {
    private int id;
    private String name;
    public Person(String n) {
        name = n;
    }
    public Person(int i, String n) {
        id = i;
        name = n;
    }
    public int getId() {
        return id;
    }
    public String getName() {
```

```
        return name;
    }
}
```

3. MySQLPersonDAO.java

```
/**
 * File : MySQLPersonDAO.java 31/05/23
 * Penulis : Devin Januar Siahaan - 24060121140158
 * Deskripsi : implementasi PersonDAO untuk MySQL
 */
import java.sql.*;
public class MySQLPersonDAO implements PersonDAO {
    public void savePerson(Person person) throws Exception {
        String name = person.getName();
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/pbo", "root",
"Delker2201");
        String query = "INSERT INTO person(name) VALUES ('" + name +
"')";
        System.out.println(query);
        Statement s = con.createStatement();
        s.executeUpdate(query);
        con.close();
    }
}
```

4. DAOManager.java

```
/**
 * File : DAOManager.java 31/05/23
 * Penulis : Devin Januar Siahaan - 24060121140158
 * Deskripsi : pengelola DAO dalam program
 */
public class DAOManager {
    private PersonDAO personDAO;
    public void setPersonDAO(PersonDAO person) {
        personDAO = person;
    }
    public PersonDAO getPersonDAO() {
        return personDAO;
    }
}
```

5. MainDAO.java

```
/**
 * File : MainDAO.java 31/05/23
 * Penulis : Devin Januar Siahaan - 24060121140158
 * Deskripsi : Main program untuk akses DAO
 */

public class MainDAO {
    public static void main(String args[]) {
        Person person = new Person("Devin");
        DAOManager m = new DAOManager();
        m.setPersonDAO(new MySQLPersonDAO());
        try {
            m.getPersonDAO().savePerson(person);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

6. Membuat database dengan nama “PBO” dan membuat tabel pada database

```
Devin<show databases;
+-----+
| Database |
+-----+
| company  |
| information_schema |
| inventory |
| mysql     |
| organisasi |
| pbo       |
| performance_schema |
| prak_8    |
| sys       |
| tugas1    |
| tugas3    |
+-----+
11 rows in set (0.01 sec)

Devin<use pbo;
Database changed
Devin<show tables;
Empty set (0.00 sec)

Devin<CREATE TABLE person(
    -> id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
    -> name VARCHAR(100));
Query OK, 0 rows affected (0.04 sec)

Devin<select * from person;
Empty set (0.01 sec)
```

Penjelasan :

Gunakan CREATE DATABASE pbo; untuk membuat database baru yaitu database pbo.

Lalu gunakan database pbo untuk membuat tabel person dengan cara use pbo; lalu CREATE TABLE person. Setelah dimasukkan atribut dari tabel tersebut, bisa dicek

terlebih dahulu apakah tabel tersebut berhasil dibuat, bisa dengan menggunakan SHOW TABLES; atau langsung mengecek isi dari tabel tersebut dengan cara SELECT * FROM person;. Karena tabel belum diisi dengan data apapun, maka output yang diberikan adalah empty set.

7. Compile kode dengan perintah: `javac *.java`

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Devin Januar Siahaan\Downloads\SEMESTER 4 ( KULIAH )\Praktikum PBO C1\Pertemuan 9> Javac *.java
PS C:\Users\Devin Januar Siahaan\Downloads\SEMESTER 4 ( KULIAH )\Praktikum PBO C1\Pertemuan 9>
```

Penjelasan :

Setelah menjalankan semua file dalam folder, ketika kita melakukan kompilasi, tidak ada output yang dihasilkan. Hal ini terjadi karena program belum terhubung dengan SQL dan belum ada data yang tersedia. Meskipun tidak ada output yang dikeluarkan, hal ini menandakan bahwa tidak ada kesalahan (error) dalam kode. Dengan demikian, meskipun program belum berinteraksi dengan basis data SQL dan tidak memiliki data, program tetap berjalan tanpa adanya masalah yang terdeteksi.

8. Jalankan MainDAO.java dengan perintah:

`java -classpath .\mysql-connector-j-8.0.33.jar;. MainDAO`

```
Devin<select * from person;
+----+-----+
| id | name |
+----+-----+
|  1 | Devin |
+----+-----+
1 row in set (0.01 sec)
```

B. Menggunakan Persistent Object sebagai Objek Terserialisasi

1. SerializePerson.java

```
/**
 * File : SerializePerson.java 31/05/23
 * Penulis : Devin Januar Siahaan - 24060121140158
 * Deskripsi : pengelola DAO dalam program
 */
import java.io.*;
class Person implements Serializable {
```

```

private String name;
public Person(String n) {
    name = n;
}
public String getName() {
    return name;
}
}
public class SerializePerson {
    public static void main(String[] args) {
        Person person = new Person("Devin");
        try {
            FileOutputStream f = new FileOutputStream("person.ser");
            ObjectOutputStream s = new ObjectOutputStream(f);
            s.writeObject(person);
            System.out.println("Selesai menulis objek person");
            s.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Hasil Keluaran :

```

[Running] cd "c:\Users\Devin Januar Siahaan\Downloads\SEMESTER 1\Koding\Java\Serialize Person"
Selesai menulis objek person

[Done] exited with code=0 in 0.799 seconds

```

2. ReadSerializePerson.java

```

/**
 * File : ReadSerializedPerson.java 31/05/23
 * Penulis : Devin Januar Siahaan - 24060121140158
 * Deskripsi : Program untuk serialisasi objek Person
 */

import java.io.*;

public class ReadSerializedPerson {
    public static void main(String[] args) {
        Person person = null;
        try {

```

```
        FileInputStream f = new FileInputStream("person.ser");
        ObjectInputStream s = new ObjectInputStream(f);
        person = (Person) s.readObject();
        s.close();
        System.out.println("serialized person name = " +
person.getName());
    } catch (Exception ioe) {
        ioe.printStackTrace();
    }
}
```

Hasil Keluaran :

```
[Done] exited with code=0 in 0.903 seconds

[Running] cd "c:\Users\Devin Januar Siahaan\Downloads\SEMESTER
serialized person name = Devin

[Done] exited with code=0 in 0.861 seconds
```